

**IN THE CLAIMS:**

1. (Currently Amended) A method for operating a data storage system, comprising:  
2. creating a writable virtual disk (vdisk) at a selected time, the writable vdisk referencing changes in data stored in the data storage system after the writable vdisk was created, the writable vdisk having a plurality of holes where each hole is at locations referencing data which has not been changed since the writable vdisk was created;  
3. maintaining a backing store, the backing store referencing data stored in the data storage system which has not been changed since the writable vdisk was created;  
4. loading blocks of the writable vdisk from a disk into a memory, the loaded blocks including a writable vdisk indirect block having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a particular hole of the a plurality of holes, where each hole instructs the data storage system to examine a corresponding virtual block number pointer in the backing store;  
5. loading blocks of the backing store from a disk into the memory, the loaded blocks including a backing store indirect block having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, one or more backing store indirect block fields having a pointer to a data block;  
6. searching each field of the writable vdisk indirect block for a hole; and  
7. replacing each field having a hole in the writable vdisk indirect block with a new pointer to the data block referenced by the corresponding backing store indirect block field to update the writable vdisk to reference both the data which is unchanged since the writable vdisk was created and the data which has been changed since the writable vdisk was created.

2. (Previously Presented) The method of claim 1, further comprising:  
3. dirtying the data block pointed to by the backing store indirect block to enable write allocation of the dirty data block without altering a data content of the data block.

1       3. (Previously Presented) The method of claim 1, further comprising:  
2              choosing a new pointer for a newly allocated data block containing an unaltered  
3              data content;  
4              setting bits in block allocation structures for the newly allocated data block; and  
5              placing the new pointer to the newly allocated data block into the field of the wri-  
6              table vdisk indirect block to replace the hole.

1       4. (Previously Presented) The method of claim 3 further comprising:  
2              freeing the dirty data block; and  
3              writing the newly allocated data block to disk.

1       5. (Previously Presented) The method of claim 4 further comprising:  
2              releasing an association of the writable vdisk to the backing store to thereby  
3              separate the writable vdisk data blocks from the backing store data blocks.

1       6. (Original) The method of claim 1 wherein the pointers contained in the writable  
2              vdisk indirect block fields and the backing store indirect block fields comprise logical  
3              volume block numbers (VBNs).

1       7. (Original) The method of claim 1 wherein the invalid pointers contained in the  
2              writable vdisk indirect block fields comprise a zero logical volume block number (VBN).

1       8. (Original) The method of claim 1 wherein the plurality of fields in the writable  
2              vdisk indirect block are a writable vdisk level 1 buffer and the plurality of fields in the  
3              backing store indirect block are a backing store level 1 buffer.

1       9. (Currently Amended) An apparatus for operating a computer database, compris-  
2              ing:  
3              a writable virtual disk (vdisk) created at a selected time, the writable vdisk refer-  
4              encing changes in data stored in the a data storage system after the writable vdisk was

5        created, the writable vdisk having a plurality of holes where each hole is at locations ref-  
6        erencing data which has not been changed since the writable vdisk was created;

7            a backing store, the backing store referencing data stored in the data storage sys-  
8        tem which has not been changed since the writable vdisk was created;

9            a backdoor message handler adapted to load blocks of the writable vdisk and  
10      backing store from disk into a memory of the storage system;

11            a writable vdisk indirect block in the memory having a plurality of fields, each  
12      field storing a valid pointer to a data block or an invalid pointer representing a particular  
13      hole of ~~the\_a~~ plurality of holes, where each hole instructs the data storage system to ex-  
14      amine a corresponding virtual block number pointer in the backing store;

15            a backing store indirect block in the memory having a plurality of fields, each  
16      backing store indirect block field corresponding to a field of the writable vdisk indirect  
17      block, each backing store indirect block field having a pointer to a data block;

18            a special loading function for searching each field of the writable vdisk indirect  
19      block for one or more fields representing a hole; and

20            a write allocator for replacing each field representing a hole in the writable vdisk  
21      indirect block with a new pointer to the data referenced by the corresponding backing  
22      store indirect block field to update the writable vdisk to reference both the data which is  
23      unchanged since the writable vdisk was created and the data which has been changed  
24      since the writable vdisk was created.

1        10. (Previously Presented) The apparatus of claim 9 wherein the write allocator fur-  
2        ther comprises:

3            a new pointer for a newly allocated data block containing an unaltered data con-  
4        tent, set bits in block allocation structures for the newly allocated data block, and place  
5        the new pointer to the newly allocated data block into the field of the writable vdisk indi-  
6        rect block to replace the hole.

1        11. (Original) The apparatus of claim 10 wherein the write allocator is further  
2        adapted to:

3 free the dirty data block and write the newly allocated data block to disk.

1 12. (Original) The apparatus of claim 9 wherein the backdoor message handler loads  
2 the blocks of the writable vdisk and the blocks of the backing store during periods of re-  
3 duced processing activity.

1 13. (Original) The apparatus of claim 9 wherein the pointers contained in the wri-  
2 table vdisk indirect block fields and the backing store indirect block fields comprise logi-  
3 cal volume block numbers (VBNs).

1 14. (Original) The apparatus of claim 9 wherein the invalid pointers contained in the  
2 writable vdisk indirect block fields comprise a zero logical volume block number (VBN).

1 15. (Original) The apparatus of claim 9 wherein the plurality of fields in the writable  
2 vdisk indirect block comprises a writable vdisk level 1 buffer and the plurality of fields in  
3 the backing store indirect block comprises a backing store level 1 buffer.

1 16-18. (Cancelled).

1 19. (Currently Amended) A data storage system apparatus, comprising:  
2 means for creating a writable virtual disk (vdisk) at a selected time, the writable  
3 vdisk referencing changes in data stored in the data storage system after the writable  
4 vdisk was created, the writable vdisk having a plurality of holes where each hole is at lo-  
5 cations referencing data which has not been changed since the writable vdisk was cre-  
6 ated;

7 means for maintaining a backing store, the backing store referencing data stored  
8 in the data storage system which has not been changed since the writable vdisk was cre-  
9 ated;

10 means for loading blocks of the writable vdisk from a disk into a memory, the  
11 loaded blocks including a writable vdisk indirect block having a plurality of fields, each

12 field storing a valid pointer to a data block or an invalid pointer representing a particular  
13 hole of ~~the-a~~ plurality of holes, where each hole instructs the data storage system to examine a corresponding virtual block number pointer in the backing store;  
14  
15 means for loading blocks of the backing store from a disk into the memory, the  
16 loaded blocks including a backing store indirect block having a plurality of fields, each  
17 backing store indirect block field corresponding to a field of the writable vdisk indirect  
18 block, one or more backing store indirect block fields having a pointer to a data block;  
19 means for searching each field of the writable vdisk indirect block for a hole; and  
20 means for replacing each field having a hole in the writable vdisk indirect block  
21 with a new pointer to the data block referenced by the corresponding backing store indi-  
22 rect block field to update the writable vdisk to reference both the data which is un-  
23 changed since the writable vdisk was created and the data which has been changed since  
24 the writable vdisk was created .

1 20. (Currently Amended) A computer readable medium, including program instruc-  
2 tions executing on a computer, the program instructions including instructions for per-  
3 forming the steps of:

4 creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-  
5 encing changes in data stored in ~~the-a~~ data storage system after the writable vdisk was  
6 created, ~~the-writable-vdisk-having-a-plurality-of-holes-where-each-hole-is-at-locations-re-~~  
7 ~~ferencing-data-which-has-not-been-changed-since-the-writable-vdisk-was-created;~~  
8 maintaining a backing store, the backing store referencing data stored in the data  
9 storage system which has not been changed since the writable vdisk was created;  
10 loading blocks of the writable vdisk from a disk into a memory, the loaded blocks  
11 including a writable vdisk indirect block having a plurality of fields, each field storing a  
12 valid pointer to a data block or an invalid pointer representing a particular hole of ~~the-a~~  
13 plurality of holes, where each hole instructs the data storage system to examine a corre-  
14 sponding virtual block number pointer in the backing store;

15 loading blocks of the backing store from a disk into the memory, the loaded  
16 blocks including a backing store indirect block having a plurality of fields, each backing

17 store indirect block field corresponding to a field of the writable vdisk indirect block, one  
18 or more backing store indirect block fields having a pointer to a data block;  
19 searching each field of the writable vdisk indirect block for a hole; and  
20 replacing each field having a hole in the writable vdisk indirect block with a new  
21 pointer to the data block referenced by the corresponding backing store indirect block  
22 field to update the writable vdisk to reference both the data which is unchanged since the  
23 writable vdisk was created and the data which has been changed since the writable vdisk  
24 was created .

1 21-22. (Cancelled).

1 23. (Currently Amended) A method for operating a data storage system, comprising:  
2 creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-  
3 encing changes in data stored in the data storage system after the writable vdisk was cre-  
4 ated, the writable vdisk having a plurality of holes where each hole instructs the storage  
5 system to examine a corresponding virtual block number pointer in a backing store~~is at a~~  
6 location referencing data which has not been changed since the writable vdisk was cre-  
7 ated;  
8 maintaining ~~a~~the backing store, the backing store referencing the data stored in  
9 the data storage system which has not been changed since the writable vdisk was created;  
10 searching each field of the writable vdisk for a hole; and  
11 referencing each hole in the writable vdisk to point to the data block referenced by  
12 the corresponding backing store indirect block to update the writable vdisk to reference  
13 both the data which is unchanged since the writable vdisk was created and the data which  
14 has been changed since the writable vdisk was created.

1 24. (Previously Presented) The method of claim 23, further comprising:  
2 dirtying the data block pointed to by the backing store indirect block to enable  
3 write allocation of the dirty data block without altering a data content of the data block.

- 1        25. (Previously Presented) The method of claim 23 further comprising:  
2                 choosing a new pointer for a newly allocated data block containing an unaltered  
3                 data content;  
4                 setting bits in block allocation structures for the newly allocated data block; and  
5                 placing the new pointer to the newly allocated data block into the field of the wri-  
6                 table vdisk indirect block to replace the hole.
  
- 1        26. (Previously Presented) The method of claim 25, further comprising:  
2                 freeing the dirty data block; and  
3                 writing the newly allocated data block to disk.
  
- 1        27. (Previously Presented) The method of claim 26 further comprising:  
2                 releasing an association of the writable vdisk to the backing store to thereby  
3                 separate the writable vdisk data blocks from the backing store data blocks.
  
- 1        28. (Previously Presented) The method of claim 23, further comprising:  
2                 including logical volume block numbers (VBNs) in the pointers contained in the  
3                 writable vdisk indirect block fields and the backing store indirect block fields.
  
- 1        29. (Previously Presented) The method of claim 23, further comprising:  
2                 using a zero logical volume block number (VBN) as the invalid pointers con-  
3                 tained in the writable vdisk indirect block fields.
  
- 1        30. (Previously Presented) The method of claim 23, further comprising:  
2                 using a writable vdisk level 1 buffer for the plurality of fields in the writable vdisk  
3                 indirect block and using a backing store level 1 buffer for the plurality of fields in the  
4                 backing store indirect block.

- 1    31. (Currently Amended) A data storage system, comprising:  
2        a writable virtual disk (vdisk) created at a selected time, the writable vdisk refer-  
3        encing changes in data stored in the data storage system after the writable vdisk was cre-  
4        ated, the writable vdisk having a plurality of holes, each hole instructing the storage sys-  
5        tem to examine a corresponding virtual block number pointer in a backing store where  
6        each hole is at a location referencing data which has not been changed since the writable  
7        vdisk was created;  
8            a backing store, the backing store referencing the data stored in the data storage  
9        system which has not been changed since the writable vdisk was created;  
10          a processor to search each field of the writable vdisk for a hole; and  
11          the processor to reference each hole in the writable vdisk to point to the data  
12        block referenced by the corresponding backing store indirect block to update the writable  
13        vdisk to reference both the data which is unchanged since the writable vdisk was created  
14        and the data which has been changed since the writable vdisk was created.
- 1    32. (Previously Presented) The system of claim 31, further comprising:  
2        the data block pointed to by the backing store indirect block are dirtied to enable  
3        write allocation of the dirty data block without altering a data content of the data block.
- 1    33. (Previously Presented) The system of claim 31 further comprising:  
2        a new pointer chosen for a newly allocated data block containing an unaltered  
3        data content;  
4        bits are set in a block allocation structures for the newly allocated data block; and  
5        a new pointer to the newly allocated data block placed into a field of the writable  
6        vdisk indirect block to replace the hole.
- 1    34. (Previously Presented) The system of claim 33, further comprising:  
2        the dirty data block is freed; and  
3        the newly allocated data block is written to disk.

- 1        35. (Previously Presented) The system of claim 34 further comprising:  
2                  an association of the writable vdisk to the backing store is released to thereby  
3                  separate the writable vdisk data blocks from the backing store data blocks.
  
- 1        36. (Previously Presented) The system of claim 31, further comprising:  
2                  logical volume block numbers (VBNs) included in the pointers contained in the  
3                  writable vdisk indirect block fields and the backing store indirect block fields.
  
- 1        37. (Previously Presented) The system of claim 31, further comprising:  
2                  a zero logical volume block number (VBN) used as the invalid pointers contained  
3                  in the writable vdisk indirect block fields.
  
- 1        38. (Previously Presented) The system of claim 31, further comprising:  
2                  a writable vdisk level 1 buffer used for the plurality of fields in the writable vdisk  
3                  indirect block and a backing store level 1 buffer used for the plurality of fields in the  
4                  backing store indirect block.
  
  
- 1        39. (Currently Amended) A computer readable media, comprising:  
2                  said computer readable media containing instructions for execution on a processor  
3                  for a method of method for operating a data storage system, the method having,  
4                  creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-  
5                  encing changes in data stored in the data storage system after the writable vdisk was cre-  
6                  ated, the writable vdisk having a plurality of holes where each hole instructs the storage  
7                  system to examine a corresponding virtual block number pointer in a backing store ~~is at a~~  
8                  location referencing data which has not been changed since the writable vdisk was cre-  
9                  ated;  
10                 maintaining ~~a~~the backing store, the backing store referencing the data stored in  
11                 the data storage system which has not been changed since the writable vdisk was created;

12        searching each field of the writable vdisk for a hole; and  
13        referencing each hole in the writable vdisk to point to the data block referenced by  
14        the corresponding backing store indirect block to update the writable vdisk to reference  
15        both the data which is unchanged since the writable vdisk was created and the data which  
16        has been changed since the writable vdisk was created.

1        40. (Currently Amended) A method for operating a data storage system, comprising:  
2            creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-  
3            encing changes in data stored in the data storage system after the writable vdisk was cre-  
4            ated, the writable vdisk having a plurality of holes where each hole instructs the data stor-  
5            age system to examine a corresponding virtual block number pointer in a backing store-  
6            where each hole is at a location referencing data which has not been changed since the  
7            writable vdisk was created;

8            maintaining ~~a-the~~ the backing store, the backing store referencing the data stored in  
9        the data storage system which has not been changed since the writable vdisk was created;  
10          searching, by a background task process, each field of the writable vdisk for a  
11        hole; and

12          referencing each hole in the writable vdisk to point to the data block referenced by  
13        the corresponding backing store indirect block to update the writable vdisk to reference  
14        both the data which is unchanged since the writable vdisk was created and the data which  
15        has been changed since the writable vdisk was created.

1        41. (Currently Amended) A data storage system, comprising:  
2            a writable virtual disk (vdisk) created at a selected time, the writable vdisk refer-  
3            encing changes in data stored in the data storage system after the writable vdisk was cre-  
4            ated, the writable vdisk having a plurality of holes where each hole instructs the data stor-  
5            age system to examine a corresponding virtual block number pointer in the backing  
6            storeis at a location referencing data which has not been changed since the writable vdisk  
7            was created;

8            a backing store, the backing store referencing the data stored in the data storage  
9       system which has not been changed since the writable vdisk was created;  
10          a background task processor to search each field of the writable vdisk for a hole;  
11       and  
12          the background task processor to reference each hole in the writable vdisk to point  
13       to the data block referenced by the corresponding backing store indirect block to update  
14       the writable vdisk to reference both the data which is unchanged since the writable vdisk  
15       was created and the data which has been changed since the writable vdisk was created.

1       42. (Currently Amended) A computer readable media, comprising:  
2            said computer readable media containing instructions for execution on a processor  
3       for a method of method for operating a data storage system, the method having,  
4            creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-  
5       encing changes in data stored in the data storage system after the writable vdisk was cre-  
6       ated, the writable vdisk having a plurality of holes where each hole instructs the data stor-  
7       age system to examine a corresponding virtual block number pointer in a backing store-  
8       where each hole is at a location referencing data which has not been changed since the  
9       writable vdisk was created;  
10          maintaining athe backing store, the backing store referencing the data stored in  
11       the data storage system which has not been changed since the writable vdisk was created;  
12          searching, by a background task process, each field of the writable vdisk for a  
13       hole; and  
14          referencing each hole in the writable vdisk to point to the data block referenced by  
15       the corresponding backing store indirect block to update the writable vdisk to reference  
16       both the data which is unchanged since the writable vdisk was created and the data which  
17       has been changed since the writable vdisk was created.